

Theorize if you want, but if the problem is right there in front of you, why not go for the facts?

 devblogs.microsoft.com/oldnewthing/20080723-00

July 23, 2008



Raymond Chen

On an internal discussion list, somebody asked a question similar to this:

My program launches a helper program. This helper program does different things based on the command line parameters, but the way I'm using it, it just prints its results to the console and exits. When I launch this program and wait for it to exit, my wait never completes. Why isn't the helper program exiting? Here's the code that I'm using to launch the helper process...

It wasn't long before people chimed in with their suggestions.

Have your main program call `exit()` at the end.

If you're redirecting stdout, you may be forgetting to drain the pipe. Otherwise, if the program generates too much output, the pipe fills and the helper program blocks writing to it.

Alas, that didn't help. Whether I redirect the output or not, the helper process still hangs.

Eventually I had to step in.

I can't believe I'm reading this discussion. It's like watching a room full of child psychologists arguing over why Billy is crying. They're all expounding on their pet theories, and none of them bothers to ask Billy, "Why are you crying?"

Connect a debugger to the helper process. See why it's crying.

You can sit around developing theories all you want, but since the problem is right there in front of you, the quickest way to figure out why the helper process isn't exiting is to connect a debugger and look at it to see why it's not exiting.

This is like the software version of the [black crayons](#) story.

Raymond Chen

Follow

